E3

for byte five, bit fifty-five through bit forty-eight for byte six and bit sixty-three through bit fifty-six for byte seven. Thus, all available bits are used in the register. This storage arrangement increases the storage efficiency of the processor. As well, with eight data elements accessed, one operation can now be performed on eight data elements simultaneously. Signed packed byte in-register representation 511 is similarly stored in a register in registers 209. Note that only the eighth bit of every byte data element is the necessary sign bit; other bits may or may not be used to indicate sign.

Please replace Table 1, on page 20, with the following:

E4

Data Format	Minimum Value	Maximum Value
Unsigned Byte	0	255
Signed Byte	-128	127
Unsigned Word	0	65535
Signed Word	-32768	32767
Unsigned Doubleword	0	$2^{32}-1$
Signed Doubleword	-2^{31}	$2^{31}-1$

Table 1

IN THE CLAIMS:

Please cancel Claims 36 and 38 without prejudice.

Presented below are the amended claims in a clean, unmarked format.

E5

15. An apparatus comprising:

a instruction decoder to receive an unpack instruction;

a first source register to hold a first packed data having a first plurality of packed data elements including a first packed data element and a third packed data element;

a second source register to hold a second packed data having a second plurality of

65

packed data elements including a second packed data element and a fourth packed data element;

a destination register to hold a third packed data;

a circuit coupled to the decoder to receive the first packed data from the first source register and the second packed data from the second source register and to unpack the first packed data and the second packed data responsive to the unpack instruction by copying the first packed data element into the destination register, copying the second packed data element into the destination register adjacent to the first packed data element, copying the third packed data element into the destination register adjacent to the second packed data element, and copying the fourth packed data element into the destination register adjacent to the third packed data element.

16. A digital processing apparatus comprising:

a decoder to receive an unpack control signal having an Intel integer opcode format comprising three or more bytes, a third byte of the three or more bytes permitting a first three-bit source register address and a second three-bit source-destination register address;

a first register to hold a first packed data having a first plurality of packed data elements including a first packed data element and a third packed data element, the first register corresponding to the first three-bit source register address;

a second register to hold a second packed data having a second plurality of packed data elements including a second packed data element and a fourth packed data element, the second register corresponding to the second three-bit source-destination register address;

a circuit to receive the first packed data from the first register and the second packed data from the second register, and in response to the unpack control signal, to copy the first packed data element into the second register, copy the second packed data

65

element into the second register adjacent to the first packed data element, copy the third packed data element into the second register adjacent to the second packed data element, and copy the fourth packed data element into the second register adjacent to the third packed data element.

17. The digital processing apparatus recited in Claim 16 wherein the decoder is further to receive the unpack control signal having an Intel integer opcode format as described in the "Pentium® Processor Family User's Manual," the Intel integer opcode format comprising three or more bytes, a first byte and a second byte of the three or more bytes permitting an operation code to specify an unpack operation interleaving low order packed byte elements, word elements or doubleword elements from the first and second packed data;

18. A computer system comprising:

a memory to hold an unpack instruction having an Intel integer opcode format comprising three or more bytes, one of the three or more bytes permitting a first three-bit source register address and a second three-bit source-destination register address;

a storage device to hold software, the software configured to supply the unpack instruction to the memory for execution;

a processor enabled to receive and decode the unpack instruction from the memory, the processor including: a first register corresponding to the first three-bit source register address to hold a first packed data having a first plurality of packed data elements including a first packed data element and a third packed data element, a second register corresponding to the second three-bit source-destination register address to hold a second packed data having a second plurality of packed data elements including a second packed data element and a fourth packed data element, and a circuit to receive the first packed data from the first register and the second packed data from the second register and to

E5

copy the first packed data element into the second register, copy the second packed data element into the second register adjacent to the first packed data element, copy the third packed data element into the second register adjacent to the second packed data element, and copy the fourth packed data element into the second register adjacent to the third packed data element.

E6

24. The apparatus of Claim 15, the unpack instruction having an Intel integer opcode format comprising three bytes, a third byte of the three bytes permitting a source register address and a source-destination register address.

25. The apparatus of Claim 24, the source register address and the source-destination register address each consisting of three bits.

26. The apparatus of Claim 24, the first source register corresponding to the source register address.

27. The apparatus of Claim 24, the second source register corresponding to the source-destination register address.

28. The apparatus of Claim 27, the destination register corresponding to the source-destination register address.

29. The apparatus of Claim 24 wherein the decoder further decodes the unpack instruction, a first byte and a second byte of the three bytes comprising an operation code specifying an unpack operation to interleave low order packed elements from the first and

second packed data, the elements selected from the group consisting of byte elements, word elements and doubleword elements.

E6

30. The apparatus of Claim 24 further comprising:

a memory to hold the unpack instruction; and

a storage device to hold software, the software configured to supply the unpack instruction to the memory for execution.

31. The apparatus of Claim 30, the instruction decoder to receive and decode the unpack instruction from the memory, the first source register corresponding to the source register address, the second source register corresponding to the source-destination register address.

32. The apparatus of Claim 31, the destination register corresponding to the source-destination register address.

33. The apparatus of Claim 32, the source register address and the source-destination register address each consisting of three bits.

35. The apparatus of Claim 15 wherein the first packed data element is a low order data element of the first packed data and the second packed data element is a low order data element of the second packed data and the unpack instruction comprises an opcode field to contain one of a set of operation codes to specify an unpack operation interleaving low order data elements from the first and the second pluralities of packed data elements, the opcode field specifying data elements selected from the group consisting of byte elements, word elements and doubleword elements.

E6

37. The apparatus of Claim 15 wherein the first packed data element is a high order data element of the first packed data and the second packed data element is a high order data element of the second packed data and the unpack instruction comprises an opcode field to contain one of a set of operation codes to specify an unpack operation interleaving high order data elements from the first and the second pluralities of packed data elements, the opcode field specifying data elements selected from the group consisting of byte elements, word elements and doubleword elements.